

The term Architecture in IT generally refers to Client Server Architecture. A client is defined as a requester of services and a server is defined as the provider of services. A single machine can be both a client and a server depending on the software configuration. There are different types of Client-Server architecture available. Some of them are 2-tier architecture, 3 tier architecture, 4 tier architecture and n tier architecture.

2-tier architecture

In 2-tier, the application logic is either buried inside the User Interface on the client or within the database on the server (or both). With two tier client/server architectures (see Two Tier Software Architectures), the user system interface is usually located in the user's desktop environment and the database management services are usually in a server that is a more powerful machine that services many clients. Processing management is split between the user system interface environment and the database management server environment. The database management server provides stored procedures and triggers

3-tier architecture

In 3-tier, the application logic (or) process lives in the middle-tier, it is separated from the data and the user interface. 3-tier systems are more scalable, robust and flexible. In addition, they can integrate data from multiple sources. In the three tier architecture, a middle tier was added between the user system interface client environment and the database management server environment. There are a variety of ways of implementing this middle tier, such as transaction processing monitors, message servers, or application servers. The middle tier can perform queuing, application execution, and database staging. For example, if the middle tier provides queuing, the client can deliver its request to the middle layer and disengage because the middle tier will access the data and return the answer to the client. In addition the middle layer adds scheduling and prioritization for work in progress. The three tier client/server architecture has been shown to improve performance for groups with a large number of users (in the thousands) and improves flexibility when compared to the two tier approach. Flexibility in partitioning can be as simple as "dragging and dropping" application code modules onto different computers in some three tier architectures. A limitation with three tier architectures is that the development environment is reportedly more difficult to use than the visually-oriented development of two tier applications. The most basic type of three tier architecture has a middle layer consisting of Transaction Processing (TP) monitor technology. The TP monitor technology is a type of message queuing, transaction scheduling, and prioritization service where the client connects to the TP monitor (middle tier) instead of the database server. The transaction is accepted by the monitor, which queues it and then takes responsibility for managing it to completion, thus freeing up the client.

4-tier Architecture

In a 4-tier architecture, all of the data storage and retrieval processes are logically and usually physically located on a single tier. A 4-tier architecture allows an unlimited number of programs to run simultaneously, send information to one another, use different protocols to communicate, and interact concurrently. This allows for a much more powerful application, providing many different services to many different clients. In this application we will have following 4-Tiers

1. Business Object [BO]
2. Business Access Layer [BAL]
3. Data Access Layer [DAL]
4. UI (4-Tier) folder [UI]